

### **REMARKS/ARGUMENTS**

Claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 remain for consideration by the Examiner. No claim amendments are made by this Amendment with the listing of claims being provided for the convenience of the Examiner.

#### **Claim Rejections under 35 U.S.C. §103**

In the final Office Action, claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. No. 6,163,781 ("Wess") in view of U.S. Pat. No. 5,899,998 ("McGauley") further in view of U.S. Pat. No. 6,363,388 ("Sprenger"). The rejection of the claims is traversed based on the following remarks.

The initial portion of the discussion presents arguments from the last three Amendments that distinguish Wess from the method of claim 1, with these arguments being presented mainly unchanged because these arguments are still relevant. Remarks are then provided that address the Response to Arguments. The discussion then turns to a review of the remarks provided in the prior Amendment that addressed McGauley's teaching, which is followed by remarks addressing the Response to Arguments relevant to McGauley. The discussion then turns to Sprenger and its teaching. Applicants respectfully believe that these remarks fully distinguish the pending claims from the combined teaching of these three references and show that these reference fail to support a rejection under 35 U.S.C. §103.

As noted in para. [0003] of the application and the prior Amendments, the invention is addressing the problems associated with prior E-commerce systems in which a database loader was used to load information but only in the form of tables. With reference to para. [0036], the invention uses an import/export utility, that knows nothing of the E-commerce system, to deliver information from an external data file to a business object that does the "real work" such as performing tasks including adding, deleting, or updating the data (see, for example, claim 35). The changes in the data can be performed without requiring the business object to be changed. As shown in Fig. 1, the import/export utility communicates with the

business object and gathers the imported file and provides a database that is used during the processing of the import file data by the business object. In this manner, the business object is able to import/export data, such as with relation to an E-commerce system, without use of a standard database loader and, hence, to load data not in the form of a table.

With this background in mind, claim 1 is directed to a method of processing data with a utility. The method includes selecting a file that includes a name of a business object, uploading the file to a server, storing file data in a database of the utility, delivering the data to the business object, and performing a task on the data with the business object. Also, the method of claim 1 calls for the business object to code to provide an interface to support an export operation and the task performed by the business object to include adding, deleting, or updating of the data delivered to the business object. The combination of Wess, McGauley, and Sprenger fail to teach or suggest all of these elements. Hence, the rejection of claim 1 is improper and should be withdrawn.

More specifically, the Office Action cites Wess at col. 6, lines 11-14, col. 6, lines 16-29 and 53-57, and col. 13, lines 3-10 for teaching selecting a file that includes a name of a business object. Applicants disagree with this construction of Wess. Wess at col. 2, lines 30-40 makes it clear that its method is addressing problems associated with having a database with many null data values, and beginning at line 6, col. 4, Wess provides a brief summary of its method that uses a table of defined variable symbols and comparing operations to try to reduce the amount of null data values in its relational databases. Hence, Wess is addressing a different problem than Applicants and uses a different technique to address that problem.

In the cited col. 6, lines 16-29, Wess is describing the converting of data received at a network interface 112 including “textually-based data objects” into formats expected by the system, but at this citation and elsewhere, Wess fails to teach selecting a file that has a business object name in the file. This named object is then used to perform processing on the data of the file, and Wess fails to suggest

that its "data objects" are business objects as defined by Applicants. For at least this reason, claim 1 is allowable over Wess.

The Response to Arguments states that "Wess teaches a file that has an object name in the file (see the Fig. 7 and associated text)" and "In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the data object is not the same as the business object as defined by the Applicant) are not recited in the rejected claim(s)." As to the first statement/response, Applicants' remarks clearly point out that at the cited col. 6, lines 16-29 Wess describes converting data received at a network interface including "textually-based data objects" into formats but fails to teach selecting a file that has a business object **NAME** in the file that is then used to perform processing on the data of the file. The Examiner appears to be trying to use the textually-based data objects as both the file that is selected or received and also teaching the business object that is later used to perform a task on that data by invoking code in the business object. Figure 7 of Wess shows an example of "five data object instances processable by the system of FIG. 1" (per col. 5, lines 42-43). These appear to be very similar to the file or image file described in Applicants' specification beginning toward the end of page 8, but fail to show a file that is selected to include a name of a business object that has code to process the data in the file. The Examiner is requested to provide a citation to Figure 7 or elsewhere in Wess where the name of a business object is included in the data object instances of Wess or to withdraw the rejection.

As to the second statement in the Response to Arguments, the business object is defined in the claim as being able to perform a task on delivered data "including invoking a code included in the business object, wherein the task comprises adding, deleting, or updating data delivered to the business object and wherein the code includes an interface to support an export operation." The Response statement seems to be indicating that Applicants were simply arguing "business" is different from "data" but Wess teaches objects. This is not the full argument with the Examiner citing the textually-based data object of Figure 7 as

showing the business objects defined in claim 1. Applicants can find no showing in Wess' Figure 7 or elsewhere that the data object instances could process data provided in a different file, the data object instances include code that can be invoked, or that the tasks of adding, deleting, or updating would be performed by such invoked code. Hence, Applicants are not arguing limitations that are not present in the claim but instead are asking for the Examiner to replace Wess with a reference that shows the type of objects as defined in claim 1 or to withdraw the rejection.

The last three Office Actions indicate that Wess fails to teach "starting a session, a business object, delivering the data to the business object corresponding to the name uploaded to the server, with the business object, performing a task on the delivered data, the task performing including invoking a code included in the business object."

Due to these deficiencies in Wess, the November 17, 2005 and April 18, 2006 Office Actions cite McGauley for teaching each of these limitations of claim 1 that are not found in Wess (except for an interface to support export generation for which Sprenger is cited). Specifically, the following discussion stresses how McGauley fails to teach or suggest the data delivering step and the performing a task on the delivered data with the business object step. Hence, in addition to Wess failing to show the file selecting step as discussed above, McGauley fails to overcome the other deficiencies of Wess.

McGauley is cited as teaching business objects (at "update objects 240; col. 8, line 60-col. 9, line 21"), delivering the data to the business object corresponding to the name uploaded to the server (at "the data model...tags attached; col. 11, lines 60-62), performing a task on the delivered data (at "The update type...new record object; col. 9, lines 46-52), and also where the task performing includes invoking a code in the business object (at "the update object...audit fields 247; col. 9, lines 14-21). Applicants disagree with this construction of McGauley.

The "updated objects" in McGauley are data files or data objects that are passed throughout a network to allow a patient's medical records to be kept within a

portable data carrier (PDC) or smart card and/or at databases at service providers (point-of-service (POS) stations). A good summary of the McGauley method is provided from col. 3, line 61 to col. 4, line 51. In this description, it can be seen that “data is transmitted via ‘update objects’” in “traditional telecommunication channels.” The “core of each update object is an element of information or an item of data that has been generated by a medical transaction at a POS station” (such as an X-ray report or a laboratory test result. “Tags” are provided in the update objects and “rule sets” are provided in the system to “guide each update object to its targeted PDC, POS and administrative databases.”

Hence, the “objects” discussed in McGauley are data objects that are passed throughout a medical data system, and McGauley teaches using tags in the objects and rules at the POS for making sure that the objects are passed to the correct or proper devices (smart cards carried by the patients and databases maintained at POS stations). However, the update objects do not teach the business objects of claim 1, and their teaching does not overcome the various shortcomings of Wess.

Specifically, the update objects and their handling and configuration (including the “tags” and the rule sets spread over a system) do not teach the method of claim 1, which includes selecting a file that includes a name of a business object (i.e., where does McGauley teach this – as it teaches passing the update objects about a system and does not teach retrieving the name of the update object from a selected file) delivering the data to the business object (the update objects are or include the data so data is not delivered to them except for modifications to update the patient records), and performing a task on the data with the business object (again, the update objects are or include the data and any processing of the data is done by an application at the POS or device from which the PDC is access the system). Further, as amended, the method of claim 1 also calls for the business object to code to provide an interface to support an export operation, which is not taught by the update objects, and the task performed by the business object include adding, deleting, or updating of the data delivered to the business object, which is not shown by the update object (which carry the data

elements but do not perform these tasks). For these reasons, McGauley does not overcome the deficiencies in Wess, and the combined teachings of these two references does not make the method of claim 1 obvious.

The Response to Arguments appears to be combining record objects, update objects, and processes that operate within the McGauley system in an attempt to find the “business objects” and the steps of the method of claim 1 related to such business objects. This argument is not persuasive for the reasons provided above. Further, the Response to Arguments asserts that McGauley does teach “performing a task on the delivered data, the task performing including invoking a code included in the business object” by teaching “the type of action [that] is needed to carry out” at col. 9, lines 46-52. However, at this citation, McGauley states “the update type identifier indicates the type of processing action that is intended when the update object reaches its destination database(s).” This does not teach that using a business object to perform a task by invoking code in the business object itself. Instead, the data of the update object can be added to a record, a record may be deleted, or a record may be deleted after the data reaches its destination database. However, there is no teaching that the code to perform such actions is provided in the update object (or record object for that matter). Further, the Response to Arguments points out that McGauley does not teach and is not cited for teaching the file selecting step. As noted above, Wess fails also to teach this step. For these reasons, Applicants still believe McGauley fails to provide the teaching for which it is cited, and, therefore, this reference when combined with Wess fail to teach the method of claim 1.

Further, when the teaching of Sprenger is added to that of Wess and McGauley, the method of claim 1 is not taught nor made obvious. Sprenger is not cited for overcoming the deficiencies of Wess and McGauley, and as a result, claim 1 is believed allowable over these three references because of the failings of Wess and McGauley. The Response to Arguments point out that Sprenger is only cited for teaching code that may include an interface to support an export generation and not the performing step. The following discussion is provided only in the sake of

completeness to show that Sprenger fails to overcome the deficiencies of Wess and McGauley.

As discussed in the previously-filed Amendments, Sprenger in its Summary and elsewhere makes it clear that its method is a data management method that uses agents and minions (see, Sprenger at col. 8, line 60 and on for a discussion of agents and minions) to perform various data management processes, but nowhere in Sprenger is it taught to perform an operation with a business object named in an import file on data uploaded to a database by a utility. At the cited portions of Sprenger, the use of objects is described but there is no teaching of a utility uploading a file and then updating the data based on the performance of the operation by the invoked business object.

For example, Sprenger at col. 5, lines 1-15 teaches that objects can be instantiated from an access layer by accessing the database and later saved to the database. Sprenger does not teach invoking a business object based on an import file and then, using the invoked business object to perform an operation on data in a utility database. For this reason, the pending claims are believed allowable over the combination of Wess and McGauley in further view of Sprenger.

Further, Sprenger at col. 14, lines 59-65 discusses operation of an "EventLogMinion" but this element does not perform validation of data and does not teach that the business object is named in the file providing the data. With this in mind, Sprenger also fails to teach the limitations of claim 1 including delivering the data to the business object and then performing a task on the data with the business object that changes the data.

Specifically, Wess fails to teach the use of business objects and at col. 14, lines 59-65, Sprenger states a minion "provides a central point of access for all advisory messages that need to be stored persistently in the database. The "events" in the log are descriptions of some event in time, such as the observed failure of a critical component, or the failure of a job stream to complete. The events stored in the log describe unusual conditions that require the attention of the user..." Applicants could find no suggestion in Sprenger of either delivering data

from to the business object or that a task that changes the data is performed on the data by the business object. For these additional reasons, claim 1 is allowable over the combined teaching of Wess, McGauley, and Sprenger.

Claims 3, 4, 6, 7, 9-13, 15-18, 21-32, and 34 depend from claim 1 and are believed allowable as depending from an allowable base claim. Further, claim 34 calls for the business object to perform validation after receiving the data, and this limitation is not shown or suggested by the cited references. In a prior Office Action in rejecting claim 14, the Examiner admitted that Wess fails to teach the business object being responsible for validating the data in the selected file but cited Sprenger at col. 14, lines 59-65 as providing this teaching. In the prior Office Action, however, the Examiner changed the construction of Wess and cited col. 8, lines 46-60 of Wess for teaching the validation with the business object. But, at this citation, Wess discusses use of a comparator comparing value attributes but as noted above this comparator is not provided in a business object to which data is delivered. Therefore, Wess fails to teach the validation with the business object. For these additional reasons, claim 34 is not shown or suggested by Wess and McGauley.

Independent claim 33 is directed to a method of loading information to a network application and includes steps of operating a utility to invoke a business object associated with a business object name from an import file. In the method of claim 33, the business object performs an operation on data delivered by the utility and the method further includes updating the data in a utility database based on the performance of the operation of the invoked business object. The operating of the utility to invoke the business object and the updating of the data steps are similar to limitations presented in claim 1, and the reasons for allowing claim 1 over the combined teaching of Wess, McGauley, and Sprenger are believed applicable to claim 33.

More specifically, Wess fails to teach receiving a selection of an import file that includes a name of a business object. This named business object is then used to perform processing on the data of the file, and Wess fails to suggest that its

"data objects" are business objects as defined by Applicants and the Examiner's prior Response to Argument in the final Office Action before filing of an RCE confirmed that Wess only teaches data objects (see, fourth paragraph of Response to Arguments). For at least this reason, claim 33 is allowable over Wess. Further, as discussed with reference to claim 1, McGauley does not overcome the deficiencies of Wess, and particularly, provides no teaching of the uploading, operating, and updating steps of claim 33 that involve the business object. Hence, Wess, McGauley, and Sprenger do not support a rejection of independent claim 33.

**Conclusions**

In view of all of the above, the claims are believed to be allowable, and it is requested that a timely Notice of Allowance be issued in this case.

No fee is believed due with this submittal. However, any additional fees associated with this submittal may be charged to Deposit Account No. 50-1123.

Respectfully submitted,



---

Kent A. Lembke, No. 44,866  
Hogan & Hartson LLP  
One Tabor Center  
1200 17th Street, Suite 1500  
Denver, Colorado 80202  
(720) 406-5378 Tel  
kalembke@hhlaw.com

June 13, 2006